We thank all the reviewers for their detailed comments and suggestions.

Review #2:

1: Thanks for pointing this connection. Yes, it is a version of the exponential weights, where the losses are the gradients of the smoothed hinge loss. We will add this remark to the revision.

2: The smoothed loss is actually quite important to our algorithm. While one could likely use an algorithm with a second-order bound like Squint to learn the weights instead of our FTRL-based approach (which we chose mainly for simplicity), we believe that the losses provided to the algorithm must be the gradients of the smoothed loss rather than just the individual correlations of the hints. The reason we think the smoothed loss is necessary (as discussed in lines 127-132) is that a linear combination of individually bad hints might produce a hint that is actually extremely good. It is not obvious how to capture this without using a loss that aggregates information from all hint sequences as our smooth loss does.

3: We agree with the suggestions and will incorporate them; thank you for the careful reading.

4, 5, 7, 8: We apologize for these typos and slight inaccuracies; we will fix them in the revision.

6: Yes, we assume $\alpha \leq 1$.

9: Our current proof is unable to show a high probability bound, but this is an interesting question, thanks! We will add this remark to the revision. We will also add the similarity of our analysis to quick-sort and paging.

Review #3:

Regarding the combiner algorithm and comparison to [Cutkosky 2019]: at first blush it is true that the [Cutkosky 2019] combiner only requires small loss at the origin, but it is actually difficult to use it in constrained settings even when appealing to the black-box constraint set reduction proposed in [Cutkosky & Orabona 2018]. This is because the reduction changes the losses in a way that might damage the regret bounds of the algorithms that are being combined. In particular, the reduction requires one to commit to a particular norm, which makes it difficult to design an algorithm that combines base algorithms that use different $p$ norms and is also constrained, as we are able to accomplish in Theorem 16. In fact, even in [Cutkosky 2019], the constrained optimism algorithm requires an ad hoc technical modification to the constraint set reduction in order to work. Nevertheless, as you suggest, we will add a short discussion about these subtleties and the limitations of the prior work.

Quantifiers: We will add the missing quantifiers to theorem statements to make them clearer. As suggested, we will add the comparator $u$ to the notation for regret in all appropriate places.

As you correctly notice, there is a small gap of $\sqrt{(\ln T)/\alpha}$ in the bounds. We will add a discussion to this effect.

Review #4:

Our main algorithm has two parts: (i) an algorithm to find an optimal combination of different hint sequences for a known value of $\alpha$, and (ii) a combiner algorithm to deal with unknown $\alpha$. To the best of our knowledge, both involve novel techniques that potentially have broader applications.

(i) Smooth hinge loss: Since a combination of multiple hint sequences can be significantly superior to any of the individual hint sequences, using the individual correlations of the hints is not sufficient. We introduce a novel smoothed hinge loss precisely to deal with this issue and show that using FTRL on these new losses helps obtain a new hint sequence that can provide regret comparable to the best combination of the original hints.

(ii) Combiner algorithm: This is a new, general way to combine $K$ online learners while obtaining regret that is as good as that of the best learner, to a factor $\log K$. The closest work we are aware of is [Cutkosky 2019, “Combining Online Learning Guarantees”], but our approach is conceptually quite different and applies in different settings (see our response to Review 3 for discussion of the differences). Furthermore, we show how the combiner can be used to obtain new results outside the setting of our problem. Appendix E contains two such applications: adapting to different norms and simultaneous Adagrad and dimension-free bounds.

In addition to adding these remarks, in the revision, we will position our algorithms and the proof techniques better with respect to related literature.